REPORT DOCUMENTATION PAGE	Form Approved OMB NO. 0704-0188

1. REPORT DATE (DD-MM-YYYY) 13-10-2017	2. REPORT TYPE Final Report	3. DATES COVERED (From - To) 15-Aug-2015 - 14-Aug-2017

4. TITLE AND SUBTITLE
Final Report: Building a Flexible Nework Infrastructure for Moving Target Defense

5a. CONTRACT NUMBER
W911NF-15-1-0508

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER
611103

6. AUTHORS

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAMES AND ADDRESSES
University of California - Riverside
200 University Office Building

Riverside, CA 92521 -0001

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS (ES)
U.S. Army Research Office
P.O. Box 12211
Research Triangle Park, NC 27709-2211

10. SPONSOR/MONITOR'S ACRONYM(S)
ARO

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
66813-CS-RIP.1

12. DISTRIBUTION AVAILIBILITY STATEMENT
Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES
The views, opinions and/or findings contained in this report are those of the author(s) and should not contrued as an official Department of the Army position, policy or decision, unless so designated by other documentation.

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	15. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON Srikanth Krishnamurthy
a. REPORT	b. ABSTRACT	c. THIS PAGE			
UU	UU	UU	UU		19b. TELEPHONE NUMBER 951-827-2348

RPPR Final Report
as of 17-Nov-2017

Agency Code:

Proposal Number: 66813CSRIP **Agreement Number: W911NF-15-1-0508**

INVESTIGATOR(S):

Name: Kadangode K. Ramakrishnan
Email: kk@cs.ucr.edu
Phone Number: 9518275639
Principal: N

Name: Srikanth Krishnamurthy
Email: krish@ucr.edu
Phone Number: 9518272348
Principal: Y

Organization: **University of California - Riverside**
Address: 200 University Office Building, Riverside, CA 925210001
Country: USA
DUNS Number: 627797426 EIN: 956006142
Report Date: 14-Nov-2017 Date Received: 13-Oct-2017
Final Report for Period Beginning 15-Aug-2015 and Ending 14-Aug-2017
Title: Building a Flexible Nework Infrastructure for Moving Target Defense
Begin Performance Period: 15-Aug-2015 **End Performance Period:** 14-Aug-2017
Report Term: 0-Other
Submitted By: Srikanth Krishnamurthy Email: krish@ucr.edu
Phone: (951) 827-2348
Distribution Statement: 1-Approved for public release; distribution is unlimited.

STEM Degrees: **STEM Participants:**

Major Goals: The goal of this project was to build a SDN testbed that supported security and networking research at UCR. Primarily it targeted efforts relating to validating solutions for moving target defense developed in several ARL/ARO projects including the cyber-security CRA.

Accomplishments: The testbed was constructed and has been instrumental in aiding several research efforts as outlined in the attached report.

Training Opportunities: The testbed was put together by graduate students Azeem Aqil, Ali Mohammad Khan and Ahmed Atya. Ahmed has since graduated and the other two students are on their way to graduation.

Results Dissemination: Nothing to Report

Honors and Awards: Nothing to Report

Protocol Activity Status:

Technology Transfer: Nothing to Report

PARTICIPANTS:

Participant Type: PD/PI
Participant: Srikanth Krishnamurthy
Person Months Worked: 1.00 **Funding Support:**
Project Contribution:
International Collaboration:
International Travel:
National Academy Member: N
Other Collaborators:

Participant Type: Co PD/PI
Participant: Kadangode Ramakrishnan
Person Months Worked: 1.00 **Funding Support:**
Project Contribution:
International Collaboration:
International Travel:
National Academy Member: N
Other Collaborators:

Final Report: Grant: W911NF1510508
Title: Building a Flexible Network Infrastructure for Moving Target Defense
PIs: Srikanth Krishnamurthy and K.K.Ramakrishnan

Testbed: The grant supported the deployment of the SDN testbed at UCR is composed of 11 powerful servers and 5 powerful, SDN enabled switches. The details of the equipment obtained from support from the grant are as follows.

Servers:

- 4 Dell servers with 28 cores and 192GB of ram
- 3 Dell servers with 22 CPU cores, 256GB of RAM, and 10G Ethernet ports
- 3 Dell servers with 20 CPU cores, 256GB of RAM, 10G Ethernet ports and a Nvidia Tesla p100 GPU
- 1 Dell server with 14 CPU cores and 128GB of RAM

Switches:

- 2 48 port SDN enabled HP switches (HP Aruba 3800 Switch)
- 2 72 port SDN enabled arista switches. These particular switches have a linux based OS installed. Which means they are a lot more configurable then simple SDN enabled switches (Arista 7280 SE-72)
- 1 52 port SDN enables, linux based Arista switch (Arista 7150S-52)

Configuration:

We use the testbed in a variety of ways. The simplest is a dedicated SDN network wherein one of the servers is configured as a SDN controller and the switches are configured to respond to OpenFlow commands from the controller.

Research Endeavors: As described in the proposal, the testbed has been used to validate research conducted by PI Krishnamurthy in the ARL sponsored Cyber-security CRA, and NSF projects of PI Ramakrishnan. The testbed will continue to be used for these endeavors – and we will build interfaces that will allow other PIs on ARL and ARO projects to access it if requested.

Network Intrusion Detection at Scale: The SDN testbed played a big role in the testing and evaluation of Jaal, our framework for ISP scale intrusion detection. Jaal is a distributed system that assigns monitoring duties to certain nodes in the network. The key design principle that we follow is to "extract" the required information from a packet stream at monitoring points spread through out the network. Specifically, the monitors process packets and create lightweight in-network packet summaries that are of drastically smaller volume, compared to raw packets. These summaries can be used to draw inferences using rules similar to those used in smaller scale NIDS (e.g., Snort). They are sent by monitors to a central inference engine which processes them and issues alerts when attacks are detected. Jaal is similar to IDS systems like Snort or Bro in that it is signature based. However, unlike Snort and Bro, Jaal is meant to be an IDS system that can be deployed on ISP-scale networks whereas snort and bro can only be used efficiently on enterprise networks. Consequently, to ascertain the effectiveness of Jaal, it was essential to test it on ISP-scale networks. ISP networks are generally composed on hundreds of backbone routers and present some obvious challenges when one wants to test a system meant to be deployed at that scale. The UCR SDN-testbed was essential in enabling testing at this scale. We do not have actual hardware switches or routers that would enable bare-bones testing; however, the testbed is powerful enough to support complicated topology emulations.

To set up more complicated topologies, as we did for Jaal, we instantiate the desired number of VM's on the servers as NF's and then connect them using Open vSwitches (also instantiated on servers) that are connected using virtual links. All of this usually orchestrated by the Ansible open source network management tool (other tools can also be used). This process allows us to create any topology configuration. For Jaal, we used publicly available ISP topologies as the blueprints for the topologies we emulated on the testbed. The work resulted in a to appear in ACM CoNEXT 2017 [1]. The work is joint with ARL Researchers.

NFVNice: NFVnice is a particularly useful scheduling framework to efficiently use the resources of COTS hardware and software. Managing Network Function (NF) service chains requires careful system resource management. NFVnice is a user space NF scheduling and service chain management framework to provide fair, efficient and dynamic resource scheduling capabilities on Network Function Virtualization (NFV) platforms. The NFVnice framework monitors load on a service chain at high frequency (1000 Hz) and employs backpressure to shed load early in the service chain, thereby preventing wasted work. Borrowing concepts such as rate proportional scheduling from hardware packet schedulers, CPU shares are computed by accounting for heterogeneous packet processing costs of NFs, I/O, and traffic arrival characteristics. By leveraging 'cgroups', a user space process scheduling abstraction exposed by the UNIX (Linux) operating system, NFVnice is capable of controlling when network functions should be scheduled. NFVnice improves NF performance by complementing the capabilities of the OS scheduler but without requiring changes to the OS's scheduling mechanisms. We developed NFVnice on the UCR SDN/NFV platform on top of OpenNetVM, our open source NFV platform. Our controlled experiments show that NFVnice provides the appropriate rate-cost proportional fair share of CPU to NFs and significantly improves NF performance (in terms of throughput and loss) by reducing wasted work across an NF chain, compared to using the default OS scheduler. NFVnice achieves this even for heterogeneous NFs with vastly different computational costs and for heterogeneous workloads. A paper on NFVnice appeared in the prestigious and highly competitive ACM Sigcomm 2017 networking conference [2].

SDN for cellular network support: The current cellular architecture and protocols are overly complex. The 'control plane' protocol includes setting up explicit tunnels for every session and exchanging a large number of packets among the different entities (mobile device, base station, the packet gateways and mobility management entity) to ensure state is exchanged in a consistent manner. This limits scalability of the cellular network. As we evolve to having to support an increasing number of users, cell-sites (e.g., 5G) and the consequent mobility as well as the incoming wave of IoT devices, a re-thinking of the architecture and the control protocols is required. We developed CleanG, a simplified software-based architecture for the Mobile Core Network (called the EPC) and a simplified control protocol for cellular networks. Network Function Virtualization is expected to enable the dynamic management of capacity in the cloud to support the core network of future cellular networks. In CleanG, we developed a simplified protocol that substantially reduces the number of control messages exchanged to support the various events, while retaining the current functionality expected from the network. CleanG, we believe will scale better and have lower latency. We have implemented CleanG on top of our OpenNetVM platform running on the UCR SDN/NFV testbed. We have published a paper in the CAN Workshop held in conjunction with the ACM CoNext 2016 conference [3].

ML implementations on OpenNetVM: A final project that has just started is to implement Machine Learning functions on the OpenNetVM platform. This is to explore in-network ML functionality that can be used in a multitude of applications, including measurement and monitoring of network traffic as well as low latency application support. We have implemented a first example of this for a home monitoring application and have published a paper in the Machine Learning and Artificial Intelligence applications in the network Workshop held in conjunction with IEEE ICNP 2017 [4].

[1] Azeem Aqil, Karim Khalil, Ahmed Atya, Evangelos Papalexakis, Srikanth V. Krishnamurthy, Trent Jaeger, K.K.Ramakrishnan, Paul Yu and Ananthram Swami, "Jaal: Towards Network Intrusion Detection at ISP Scale," ACM CoNEXT 2017 (to appear).

[2] Sameer Kulkarni, Wei Zhang, Jinho Hwang, Shriram Rajagopalan, K. K. Ramakrishnan, Timothy Wood, Mayutan Arumaithurai, Xiaoming Fu, “NFVnice: Dynamic Backpressure and Scheduling for NFV Service Chains”, Proc. of ACM Sigcomm 2017 conference, Aug. 2017.

[3] Ali Mohammadkhan, K. K. Ramakrishnan, Ashok Sunder Rajan, Christian Maciocco, “CleanG: A Clean-Slate EPC Architecture and Control Plane Protocol for Next Generation Cellular Networks”, Proc. of Cloud-Assisted Networking 2016 Workshop at ACM CoNext’16, Dec. 2016.

[4] Aditya Dhakal, K. K. Ramakrishnan, “Machine Learning at the Network Edge for Automated Home Intrusion Monitoring”, Workshop on Machine Learning and Artificial Intelligence in Computer Networks (ML&AI @ Network), ICNP 2017.